AMENDMENTS TO THE CLAIMS

1. (Currently Amended) An apparatus for handling SSL traffic comprising an SSL proxy operable to receive a plurality of packets ~~each including an encrypted portion~~, and to examine the header of the plurality of packets to determine if the packet is an encrypted packet, the SSL proxy operable to buffer the encrypted packets until a predetermined number of encrypted packets greater than one encrypted packet are received, the SSL proxy further operable to decrypt ~~the~~ an encrypted portion of each ~~received~~ encrypted packet to form decrypted packets and forward the decrypted packets to a predetermined destination.

2. (Original) The apparatus of claim 1, wherein the SSL proxy includes a database operable to track information regarding a type of encryption scheme used to encrypt the encrypted portion.

3. (Currently Amended) The apparatus of claim 1, wherein the encrypted portion of the encrypted packets are decrypted when received and the SSL proxy buffers the received packets out of order.

4. (Original) The apparatus of claim 1, wherein the SSL proxy tracks a message authentication code used to authenticate a message.

5. (Currently Amended) The apparatus of claim 1, wherein the plurality of packets are sent by a client computer running a web browser and received by a server computer running a web server.

6. (Original) The apparatus of claim 5, wherein the SSL proxy is operable to receive unencrypted data from the server computer, encrypt the unencrypted data, and send the encrypted data to a client computer.

7. (Currently Amended) The apparatus of claim 1, wherein the SSL proxy performs encryption and decryption on packets using a single end-to-end TCP connection between a client

computer and a server and the source and destination address of the encrypted packets are unaltered.

8. (Currently Amended) A system for handling SSL traffic comprising:

a client computer running a web server ~~computer~~ operable to initiate an SSL session and to send data packets ~~with encrypted payloads~~;

a server computer running a web browser ~~computer~~ operable to support communications with the client computer; and

a SSL proxy coupling the client computer and the server computer, the SSL proxy operable to receive the data packets and to determine if the data packets are encrypted packets by examining the header of the data packets, the SSL proxy further ~~and~~ operable to decrypt ~~the~~ each encrypted payloads of ~~each~~ the encrypted packets to form decrypted packets and forward the decrypted packets to the server computer.

9. (Original) The system of claim 8, wherein the SSL proxy includes a database operable to track information regarding a type of encryption scheme used to encrypt the encrypted payloads.

10. (Currently Amended) The system of claim 8, wherein the encrypted packets are decrypted when received by the SSL proxy and the SSL proxy buffers the ~~received~~ encrypted packets out of order.

11. (Original) The apparatus of claim 8, wherein the SSL proxy tracks a message authentication code used to authenticate a message.

12. (Original) The system of claim 8, wherein the SSL proxy is operable to encrypt packets sent from the server computer to the client computer.

13. (Currently Amended) The system of claim 8, wherein a single end-to-end TCP connection exists between the client computer and the server computer and the source and destination address of the encrypted packets are unaltered.

14. (Currently Amended) The system of claim 8, wherein the SSL proxy buffers the encrypted packets until a predetermined number of packets arrive, then decrypts packets, and forwards the decrypted packets to the server.

15. (Currently Amended) A method for processing SSL packets comprising:

initializing an SSL session between a client computer and a SSL proxy;

receiving a packet including an encrypted portion at the SSL proxy;

determining if the received packet is a SSL packet by examining the header of the second packet;

placing the ~~received~~ SSL packet in a hold queue;

checking the hold queue to determine if all SSL packets expected for a given record have arrived ~~for a complete set of packets~~;

decrypting the encrypted portion of each SSL packet once all the encrypted packets expected for the given record have arrived to form decrypted packets ~~the complete set of packets are received~~; and

outputting the decrypted packets to a server computer.

16. (Currently Amended) The method of claim 15, wherein a message authentication code is checked to verify authenticity of the SSL packet set.

17. (Original) The method of claim 15, wherein non SSL packets are sent directly to the server.

18. (Currently Amended) The method of claim 15, wherein the step of placing the SSL packets in a hold queue comprises:

placing SSL packets received out of order in a queue;

decrypting SSL packets received in order and forwarding the decrypted SSL packets to a server computer;

checking the hold queue to determine if the SSL packet in the queue is next in order ~~sequence~~;

releasing the SSL packet from the hold queue if the SSL packet in the queue is the next in order ~~sequence~~; and

getting a new SSL packet if the SSL packet in the hold queue is not the next in order ~~sequence~~.

19. (Original) The method of claim 15, wherein the step of initializing further comprises initializing a single end-to-end TCP connection between the client computer and the server computer.

20. (Original) The method of claim 15, further comprising:

receiving packets with unencrypted data at a SSL proxy from the server computer;

encrypting the packets at the SSL proxy; and

sending the encrypted packets to the client computer.

21. (Currently Amended) An apparatus for decrypting network data traffic comprising a proxy operable to:

(i) receive packets addressed to a server computer, ~~the packets including an encrypted portion, a destination address, and a source address~~;

(ii) examining the header of the received to determine if the received packet is an encrypted packet;

(~~ii~~ iii) decrypt ~~the~~ an encrypted portions of the ~~received~~ encrypted packets; and

(~~iii~~ iv) send the decrypted portions to ~~a~~ the server computer without altering the destination or source address of the received packets.

22. (Original) The apparatus of claim 21, wherein the proxy is further operable to:

(i) receive packets addressed to a client computer, the packets including an unencrypted portion, a destination address, and a source address;

(ii) encrypt the unencrypted portion of the received packets; and

(iii) send the encrypted packets to the client computer without altering the destination or source address of the packets.